



Praktikum zur Lehrveranstaltung  
XML-Technologien:  
**Übung 01** (Vorübung)

Wiederholung einiger  
wichtiger Unix-Kommandos,  
**Unicode** u.a. Zeichensätze,  
**UTF-8** Codierung



# Unix-Vorübungen

Kommandos und Konzepte, die  
Sie beherrschen – oder dringend  
wiederholen – sollten



- Dateisystem
  - *inodes*, Verzeichniseinträge,
  - *Links (hard & soft)*
- Kernel, Shell
  - Speziell: Die bash
    - Eingestellt als *default*-Shell?!
    - Befehlszeilenpuffer, -vervollständigung
- Prozesse
  - pid, Scheduler, Priorisierung; Kommandos dazu
  - Vorder- und Hintergrundprozesse
- I/O
  - stdin, stdout, stderr; /dev, „mounten“



- Das Prinzip „Unix-Werkzeugkasten“

- *Pipes* bilden:

- stdout von Prozess 1 wird stdin von Prozess 2

- Ausgaben umlenken

- Beispiele:

- 1) `cat file1 | grep pattern-a | wc > resultfile`

- 2) `find ~ -name \*.zip 2> /dev/null`

- *Patterns*, reguläre Ausdrücke

- `rm *` # ☹️

- `ls a* .??? b[1236-9]cd`



- Grundlage zum Wiederholen:
  - Übungen aus dem Praktikum zur Einführung in die Informatik.
- Wichtige Kommandos:
  - `man, info`
  - `ls, cd, pwd`
  - `mkdir, rmdir`
  - `cp, mv, rm, ln`
  - `cat, more, head, tail`
  - `grep, find`
  - `ps, pstree, kill, nice, time, fg, bg`
  - `mount, umount`



- Weitere wichtige Kommandos
  - `chmod`, `chgrp`; `touch`
  - `gzip`, `gunzip`, `gzcat`; `zip`, `unzip`; `tar`
- Anwendungen, Editoren
  - `vi`, `emacs`
  - `acroread`
  - `Mozilla`, `firefox`
  
  - Speziell für XML:  
`nsgmls` (später mehr)
- Zum Nachlesen:  
SelfLinux, insb. Kapitel „Grundlagen“  
<http://www.selflinux.org/selflinux/>



# Unicode

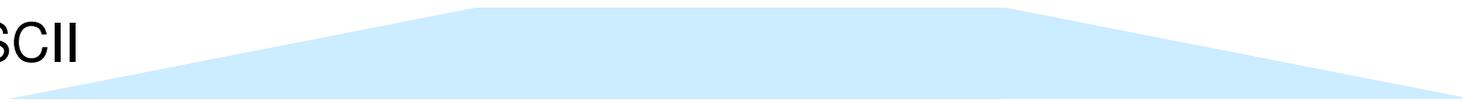
... und andere Zeichensätze



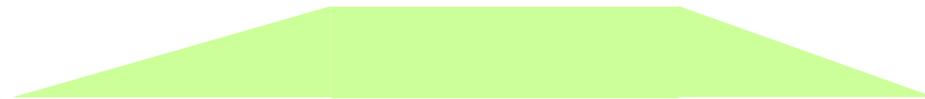
Proprietäre Zeichensätze



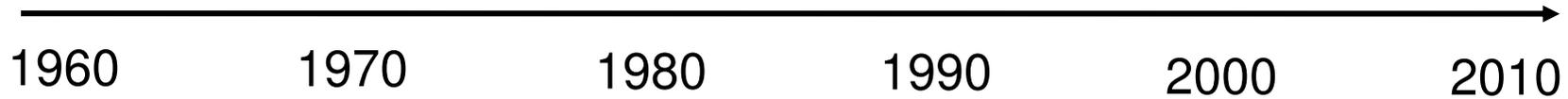
US-ASCII



ISO-8859-x



Unicode, incl. UTF-8, UTF-16





- Informationen:

- <http://czyborra.com/> War mal offline, Ersatz+mehr:  
<http://www.i18nguy.com/unicode/codepages.html>  
zu Zeichensätzen allgemein
- <http://www.unicode.org/>  
Speziell zu Unicode

- Beispiel: Buchstabe „ü“

- Codepage 437 (DOS): 0x81
- ISO-8859-1: 0xFC
- Unicode (composite): U+00FC
- Unicode (combining): U+0075, U+0308
- Unicode, UTF-8 (s.u.): U+00FC = 0xC3, 0xBC



# Unicode-Zeichenarten

---



- Basiszeichen
  - Unser normales Verständnis eines Zeichens
- Ideographische Zeichen
  - z.B. fernöstliche wie Kanji-Zeichen
- *combining characters*
  - „Pünktchen“, Akzentzeichen u.a.
  - Sie ergeben zusammen mit ihrem jeweiligen Vorläuferzeichen in einem String das endgültige Symbol
  - Beispiel: à = a`
  - Diese Zeichenkombinationen ergänzen die bereits vorhandenen Spezialzeichen
  - Die Kombinationsmethode schafft mit relativ wenigen Unicode-Einträgen eine große Vielfalt an möglichen Symbolen.
- *extenders*
  - (Unicode-Spezialthema, hier nicht behandelt)



# Unicode: Codierungen



- UCS-4:
  - Die allgemeine 4-Byte-Angabe: `U+dddddddd`
- UTF-8, UTF-16, UTF-32
- Unterscheidung im Fall UTF-16:
  - *high-endian* vs. *low-endian* mittels Sonderzeichen `xFEFF`

- UTF-8 Codierung:

<code>U+00000000</code>	–	<code>U+0000007F</code>	<code>0xxxxxxx</code>	
<code>U+00000080</code>	–	<code>U+000007FF</code>	<code>110xxxxx</code>	<code>10xxxxxx</code>
<code>U+00000800</code>	–	<code>U+0000FFFF</code>	<code>1110xxxx</code>	<code>10xxxxxx</code> <code>10xxxxxx</code>
<code>U+00010000</code>	–	<code>U+001FFFFF</code>	<code>11110xxx</code>	<code>(10xxxxxx)</code> <sub>3</sub>
<code>U+00200000</code>	–	<code>U+03FFFFFF</code>	<code>111110xx</code>	<code>(10xxxxxx)</code> <sub>4</sub>
<code>U+04000000</code>	–	<code>U+7FFFFFFF</code>	<code>1111110x</code>	<code>(10xxxxxx)</code> <sub>5</sub>

Folgebyte

Anzahl Folgebytes

- 1 bis 6 Oktetts pro Unicode-Zeichen (31 bits), niemals `xFE` oder `xFF`.
- Stets klar, ob Folgebyte vorliegt und wie viele Folgebytes notwendig!



- **Aufgabe:**
  - **Ermitteln Sie die Codes der umseitig folgenden Zeichen**
- **Hinweise:**
  - Dokumentieren Sie ihre Ergebnisse tabellarisch in Datei **xmltech-01a.txt** – diese werden noch benötigt.
  - Verwenden Sie die angegebenen Internetquellen.
  - Geben Sie stets den Unicode an.
  - Geben Sie den Code aus einer der ISO-8859-Tabellen an, incl. der Tabellenummer selbst, sofern ein ISO-Code für das Zeichen existiert.



# Suchaufgaben

---



- A) Westliche Sonderzeichen
  - Ä, ä, Ö, ö, Ü, ü; ß
- B) Währungszeichen
  - British Pounds, Euro: £, €
- C) Mathematische Sonderzeichen
  - Quantoren: „Für alle“, „es existiert“, „es existiert nicht“:  $\forall$ ,  $\exists$ ,  $\nexists$
  - „daraus folgt“, „ist äquivalent“, „ist gleich“, „ist ungleich“:  $\Rightarrow$ ,  $\Leftrightarrow$ ,  $=$ ,  $\neq$
  - Sonstiges: „ist Element von“; „alpha“, „beta“, „gamma“, das Gradzeichen (wie in: 37 °C):  $\in$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $^\circ$
- D\*) Kanji
  - Schreiben Sie „japanisch“ - auf japanisch (Ni-Hon-Go): 日本語



- UFT-8 Codes berechnen:  
Ermitteln Sie die UTF-8 Codes (Oktett-Sequenzen) der folgenden drei Unicode-Zeichen:
  - A)           β
  - B)           €
  - C\*)         ☀ („Ni“, Kanji-Zeichen für "Sonne")
- Abgabe
  - Tragen Sie Ihre Ergebnisse ein in Datei **xmltech-01b.txt**
- Hinweis zu beiden Abgabeteilen:  
Die mit \*) gekennzeichneten Aufgabenteile sind **freiwillig**



- Geben Sie folgende 2 Dateien ab:
  - **xmltech-01a.txt**
  - **xmltech-01b.txt**
- Abgaberegeln:
  - Gemäß der allgemeinen Abgaberegeln (beschrieben in eigener PDF-Datei).
- **Hinweise:**
  - Immer Name, Vorname, MatNr, etc. in der Datei angeben
  - Keine Teamlösungen - "Kopien" erhalten keine Punkte!
  - Verwenden Sie "cp", keinen Filemanager oder GUI-Tools!